

(51) Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 12/08	3 1 0 A	7608-5B		
9/46	3 4 0 F	8120-5B		
12/00	5 7 2	9366-5B		

審査請求 未請求 請求項の数 2 O L (全 15 頁)

(21) 出願番号 特願平5-55277

(22) 出願日 平成5年(1993)3月16日

(71) 出願人 000004226

日本電信電話株式会社

東京都千代田区内幸町一丁目1番6号

(72) 発明者 塩澤 恒道

東京都千代田区内幸町1丁目1番6号 日

本電信電話株式会社内

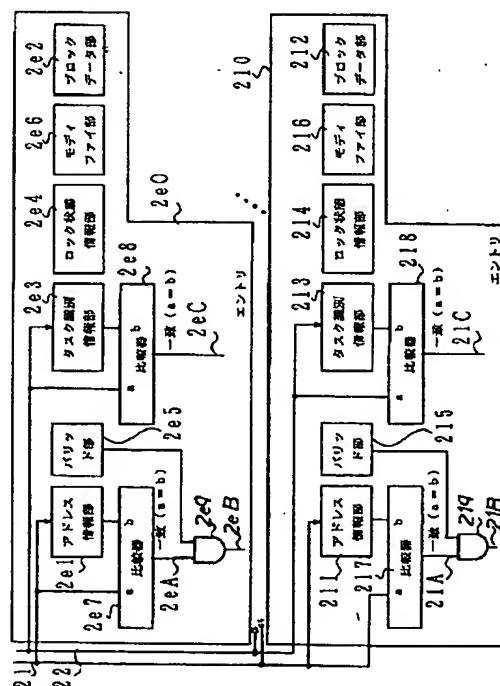
(74) 代理人 弁理士 磯村 雅俊

(54) 【発明の名称】 キャッシュメモリを用いた排他制御システム

(57) 【要約】

【目的】 マルチタスク処理における排他制御を効率良く行ない、共通データの無矛盾性の保証性能と、処理ユニットの実効的な性能とを向上させる。

【構成】 メインメモリのブロック単位のデータを各エントリに格納し、このエントリを介して、マルチタスク処理を行なう処理ユニットからのアクセスに対応するデータの転送制御と排他制御を行なうキャッシュメモリにおいて、エントリに格納されているブロックを排他制御の対象として設定した処理ユニットの各タスクの識別情報を登録するタスク識別情報登録部を、エントリごとに設ける構成とし、タスク単位で、エントリに格納されているブロックの排他制御、および、この排他制御の設定と解除を行なう。



【特許請求の範囲】

【請求項1】 タスクを分割して並列に処理する処理ユニットがアクセスするメインメモリに格納されているブロック単位のデータを、各エントリに格納し、該エントリを介して、上記処理ユニットからのアクセスに対するデータの転送制御を行なうと共に、上記処理ユニットで並列に処理される複数のタスクからの同一のエントリへのアクセスを制御してデータの矛盾を回避する排他制御を行なうキャッシュメモリにおいて、上記排他制御のために、上記エントリに、上記処理ユニットで並列に処理される各タスクの識別情報を登録するタスク識別情報登録手段と、上記エントリに格納されているブロックに対する排他制御情報を登録する排他制御情報登録手段を、上記エントリごとに設け、タスク単位で、上記エントリに格納されているブロックの排他制御、および、該排他制御の設定と解除を行なうことを特徴とするキャッシュメモリを用いた排他制御システム。

【請求項2】 請求項1に記載のキャッシュメモリを用いた排他制御システムにおいて、上記キャッシュメモリは、上記処理ユニットのタスクからの上記排他制御の解除指示に基づき、該タスクの識別情報を登録しているエントリに対する上記タスク識別情報の解除、および、該エントリに対する全てのタスク識別情報が解除された場合での上記排他制御情報の解除を行なうことを特徴とするキャッシュメモリを用いた排他制御システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、複数のタスクを並列に実行するマルチタスク処理を行なう処理ユニットからのアクセスに対するデータの転送制御を行なうキャッシュメモリを用いた排他制御技術に係わり、特に、複数のタスク間でアクセスされるデータの無矛盾性の保証を、効率良く行なうのに好適なキャッシュメモリを用いた排他制御システムに関するものである。

【0002】

【従来の技術】 一般に、メインメモリのデータ転送速度は、処理ユニットの高速な処理速度に比べて低速であり、このメインメモリの低速なデータ転送動作は、システムの性能低下の原因となっている。このような問題を解決するためには、例えば、社団法人電子情報通信学会編「電子情報通信ハンドブック」（1988年、株式会社オーム社発行）の第1670頁から第1671頁に記載のように、キャッシュメモリ（バッファ記憶）を設けることが有効である。

【0003】 このようなキャッシュメモリを介して、マルチタスク処理を行なう処理ユニットの各タスクが、共通にアクセスする共通データの読み出し、および、書き換えを行なう場合、他のタスクとの間に矛盾が発生する。例えば、タスクAとタスクBが、メインメモリのアドレスX（キャッシュメモリのエントリX）に格納して

いる値「x」に、それぞれ、「10」および「100」を加算する場合、タスクAとタスクBの処理が矛盾なく行なわれると、アドレスXの値は「x+110」となる。しかし、タスクAが、アドレスXの値「x」を読み出してからアドレスXに値「x+10」を書き込む前に、処理ユニットで実行されるタスクが、タスクBに切り替わると、タスクBが、アドレスXの値「x」を読み出し、タスクBは、アドレスXを「x+100」に書き換える。そして、タスクAが再開されると、タスクAは、アドレスXを「x+10」に書き換え、最終的なアドレスXの値は「x+10」となる。

【0004】 このような矛盾が発生しないように、各タスクは、共通データにアクセスする前に、共通データへのアクセス権の獲得を行なう。このアクセス権の獲得が成功すると、タスクは、共通データへのアクセスを行ない、共通データへのアクセス終了後に、このアクセス権を解放する。従来技術では、このアクセス権は、共通データごとに設けられ、同時には1つのタスクにのみ、アクセス権が与えられるように制御される。

【0005】 このように、従来のマルチタスクでのデータの矛盾を保証する制御においては、各タスクは、共通データにアクセスする前にアクセス権を獲得し、他のタスクからの共通データへのアクセスを抑止する制御や、あるタスクが共通データにアクセスしている間には、処理ユニットで実行中のタスクの切り替えを抑止する制御などの排他制御が必要である。一般に、このような排他制御では、対象となるデータを以下の3つの状態で管理する。

(a) リードロック状態：1つ以上のタスクによって読み出し中であり、他タスクからの書き換えアクセスを抑止する必要がある。

(b) ライトロック状態：1つのタスクによって書き換え中であり、他タスクからの読み出し、および、書き換えアクセスを抑止する必要がある。

(c) アンロック状態：何れのタスクからも全てのアクセスが可能である。

【0006】 しかし、このような排他制御技術では、各タスクが同時並列にアクセスする共通データごとに、アクセス権の確保と解放の処理を行なう必要があるため、このための処理量が増加する欠点があった。このような問題を解決するための従来技術として、同時並列にアクセスしないデータと共通データを厳密に分けずに、アクセス権の確保と解放を、複数の共通データに対して一括して行なうことにより、アクセス権の確保と解放の回数を減らす技術がある。

【0007】 しかし、このように、複数の共通データに対して一括してアクセス権の確保と解放を行なうと、排他制御の対象となる共通データの対象が増加し、無用な排他制御が行なわれるため、複数のタスクで並列に行なう処理の実行が阻害される確率が大きくなってしま

また、タスクの切り替えを抑止する排他制御では、共通データへのアクセスを頻繁に行なうタスクの実行が優先されることとなり、例えば、タスクによっては、開始から終了まで切り替えが行なわれなくなり、タスクを相互に切り替えて分割処理することによって実行の並列化を図るマルチタスクの制御が複雑になってしまう。

【0008】

【発明が解決しようとする課題】解決しようとする問題点は、複数の共通データに対して一括してアクセス権の確保と解放を行なう従来の排他制御技術では、無用な排他制御が行なわれ、マルチタスク処理が阻害される確率が大きくなってしまい、また、タスクの切り替えを抑止する従来の排他制御技術では、共通データへのアクセスを頻繁に行なうタスクの実行が優先され、マルチタスクの並列処理制御が複雑になってしまい、排他制御を効率良く行なうことができない点である。本発明の目的は、これら従来技術の課題を解決し、キャッシュメモリのエン트리への各タスクのアクセスに対する排他制御を効率良く行ない、マルチタスク処理を行なう処理ユニットの実効的な性能を向上させることを可能とするキャッシュメモリを用いた排他制御システムを提供することである。

【0009】

【課題を解決するための手段】上記目的を達成するため、本発明のキャッシュメモリを用いた排他制御システムは、(1)タスクを分割して並列に処理する処理ユニットがアクセスするメインメモリに格納されているブロック単位のデータを、各エントリに格納し、このエントリを介して、処理ユニットからのアクセスに対するデータの転送制御を行なうと共に、処理ユニットで並列に処理される複数のタスクからの同一のエントリへのアクセスを制御してデータの矛盾を回避する排他制御を行なうキャッシュメモリにおいて、排他制御のために、エントリに、処理ユニットで並列に処理される各タスクの識別情報を登録するタスク識別情報登録部と、エントリに格納されているブロックに対する排他制御情報を登録する排他制御情報登録部を、エントリごとに設け、タスク単位で、エントリに格納されているブロックの排他制御、および、この排他制御の設定と解除を行なうことを特徴とする。また、(2)上記(1)に記載のキャッシュメモリを用いた排他制御システムにおいて、キャッシュメモリは、処理ユニットのタスクからの排他制御の解除指示に基づき、このタスクの識別情報を登録しているエントリに対するタスク識別情報の解除、および、このエントリに対する全てのタスク識別情報が解除された場合での排他制御情報の解除を行なうことを特徴とする。

【0010】

【作用】本発明においては、マルチタスク処理を行なう処理ユニットの各タスクの、キャッシュメモリの同一エントリに対する排他制御を、各タスク単位で行なう。例

えば、キャッシュメモリの各エントリは、各エントリを識別するアドレス情報や、他のタスクからの全てまたは一部のアクセスを抑止する排他制御が設定された状態を示すロック状態情報などと共に、排他制御の設定(ロック)を指示したタスクの識別情報を登録する。そして、ロック状態としたエントリに対する抑止が必要なアクセスが発生した場合、このアクセスを行なったタスクを判別する。もし、他のタスクからのアクセスであれば、このタスク、もしくは、このエントリをロックしたタスクのいずれかを中断、または、無力化(中止)することにより、排他制御を行なう。このことにより、各タスクが並列にアクセスする共通なエントリのアクセス権の確保と解放の処理を行なうための処理量の増加を回避することができ、かつ、タスクの切り替えを抑止する制御も不要となる。

【0011】

【実施例】以下、本発明の実施例を、図面により詳細に説明する。図1は、本発明の排他制御システムに用いるキャッシュメモリの本発明に係わる構成の一実施例を示すブロック図であり、図2は、このキャッシュメモリを用いた情報処理システムの構成の一具体例を示すブロック図である。図2において、1は、分割した複数のタスクを並列に実行してマルチタスク処理を行なう処理ユニット、2は、本発明に係わる排他制御動作を行なうキャッシュメモリ、3は、処理ユニット1とキャッシュメモリ2間を接続するキャッシュバス、4は、処理ユニット1からのアクセスの対象となる元のデータを格納するメインメモリ、5は、キャッシュメモリ2とメインメモリ4間を接続するメモリバスである。

【0012】キャッシュバス3は、処理ユニット1から送出されるアドレスを伝達するためのアドレスバス31と、処理ユニット1とキャッシュメモリ2の間で授受されるデータを伝達するためのデータバス32と、処理ユニット1から送出されるアクセスの内容(読み出し/書き換え)を伝達するためのリード/ライトバス33と、本発明に係わり、処理ユニット1で実行中の各タスクの識別情報を伝達するためのタスク識別情報バス34と、処理ユニット1から送出されるロック(排他制御設定)指示情報を伝達するためのロック指示バス35と、キャッシュメモリ2から送出される処理ユニット1からのアクセスに対応する制御結果を伝達するための転送応答バス36と、キャッシュメモリ2から送出される処理ユニット1からのアクセスが矛盾した結果を伝達するための矛盾応答バス37とからなる。

【0013】処理ユニット1は、アドレスaで指定されるデータの読み出し、または、書き換えアクセスを行なうとき、アドレスバス31にアドレスaを出力し、リード/ライトバス33に、「読み出し」または「書き換え」の種別を出力し、タスク識別情報バス34に、処理ユニット1で実行中のタスクに予め割り当てられたタス

ク識別情報を出し、ロック指示バス35に、「ロック設定指示」または「ロック解除指示」を出し、書き換えの場合には、データバス32に、書き換えデータを出し、出力する。

【0014】本実施例においては、タスク識別情報バス34は、複数の信号線からなり、各信号線には、処理ユニット1で並列して実行する各タスクが、重複しないように割り当てられている。そして、処理ユニット1で実行中のタスクは、キャッシュメモリ2にアクセスする際に、割り当てられている信号線のみ、値「1」を出力し、他の信号線には、値「0」を出力する。また、ロック情報の解除を指示するためのバスは設けず、処理ユニット1からのアクセス時に、ロック指示バス35にロック指示が出力されていない場合に、当該するエントリに対するロック情報の解除を指示する。また、メインメモリ4は、同じ大きさのブロックに分割され、キャッシュメモリ2との間では、メモリバス5を介して、ブロック単位にメモリ内容の転送を行なう。本発明に係わるキャッシュメモリ2は、図1で示すように、メインメモリ4の各ブロック単位のメモリ内容を格納する複数のエントリからなる。

【0015】図1において、キャッシュメモリ2のe個の各エントリ210~2e0は、転送されたブロックのアドレス情報を格納するアドレス情報部211~2e1と、ブロック単位のメモリ内容を格納するブロックデータ部212~2e2と、図2のタスク識別情報バス34の信号線と同一のビット数からなり、ロック指示を行なったタスクが値「1」を出力した信号線に対応するビットに、値「1」を設定する本発明のタスク識別情報登録部としてのタスク識別情報部213~2e3と、ロック（排他制御）状態を示す本発明の排他制御情報登録部としてのロック状態情報部214~2e4と、各エントリ210~2e0に格納されている内容が有効であるか無効であるかを示すバリッド部215~2e5と、ブロックデータ部212~2e2の内容のみを書き換えたことを示すモディファイ部216~2e6と、アドレス情報部211~2e1と信号線21との内容とを比較する比較器217~2e7と、タスク識別情報部213~2e3と信号線22との内容とを比較する比較器218~2e8と、比較器217~2e7から「一致」した旨の信号が出力され、かつ、バリッド部215~2e5の内容が、「1（有効）」である場合に、信号線21B~2eBに「オン」を出力するアンドゲート219~2e9とからなる。

【0016】以下、このような構成の本実施例のキャッシュメモリの本発明に係わる動作を、図2のメインメモリ4のアドレスaで指定されるデータを含むブロックをBaとし、図2の処理ユニット1が、ブロックBaをアクセスする場合を例として、図3~図8を用いて説明する。

【0017】図3~図8は、図2における情報処理システムの本発明に係わる処理動作の第1の具体例を示すフローチャートである。本例は、主に、図2の処理ユニット1からのアクセスに対する図1における本実施例のキャッシュメモリ2の処理を示すものであり、論理回路を用いることにより、この処理の各部分を並列に実行することができる。まず、図3における処理動作の説明を行なう。図2の処理ユニット1が、図2のキャッシュバス3を介して図2のメインメモリ4のa番地のブロックBaにアクセスすると（ステップ301）、図1におけるキャッシュメモリ2は、図2のアドレスバス31に出力されているアドレスaのアドレス情報を、図1の信号線21に入力し、また、図2のタスク識別情報バス34に出力されているタスク識別情報を、図1の信号線22に入力する（ステップ302）。そして、それぞれ入力したアドレス情報とタスク識別情報とに対応して、以下のように処理を行なう。

【0018】図2の処理ユニット1から、図2のアドレスバス31を介して要求されたアドレスaのデータを含むブロックBaを格納する有効なエントリが、図1のキャッシュメモリ2に存在しない、すなわち、図1の信号線21B~2eBの出力が「オン」であるエントリが存在しない場合（ステップ303）、図1のキャッシュメモリ2は、図1のロック状態情報部214~2e4が、「0（アンロック状態）」であるエントリから、予め定められた手続きによって、エントリを選択する（ステップ304）。以下、この選択したエントリをエントリEと記述し、このエントリEは、アドレス情報部2x1、ブロックデータ部2x2、タスク識別情報部2x3、・・・で構成されているものとする。

【0019】エントリEのバリッド部2x5が「1（有効）」、かつ、モディファイ部2x6が「1（ブロックデータ部のBaとメインメモリのBaが不一致）」であれば（ステップ305）、図1のキャッシュメモリ2は、図2のメモリバス5を介して、エントリEのアドレス情報部2x1で指定される図2メインメモリ4のブロックを、ブロックデータ部2x2で書き換える（ステップ306）。そして、図1のキャッシュメモリ2は、図2のメモリバス5を介して、図2のメインメモリ4からブロックBaを読み出し、エントリEのブロックデータ部2x2に格納し、アドレスaのアドレス情報を、アドレス情報部2x1に格納し、バリッド部2x5を「1（有効）」として（ステップ307）、次の図4に示す処理を行なう。

【0020】以下、図4における処理動作の説明を行なう。図2の処理ユニット1が、図2のロック指示バス35に、ロック指示信号を出力していない場合には（ステップ308）、タスク識別情報部2x3の全てのビットに「0」を設定し（ステップ309）、ロック状態情報部2x4に、「0（アンロック状態）」を設定する（ス

テップ310)。また、ステップ308において、図2の処理ユニット1が、図2のロック指示バス35に、ロック指示信号を出力している場合には、タスク識別情報部2x3に、図2のタスク識別情報バス35に出力されている値を設定する(ステップ311)。そして、図2のリード/ライトバス33に、アクセス種別として、「書き換え」ではなく「読み出し」が出力されていれば(ステップ312)、ロック状態情報部2x4に、「1(リードロック状態)」を設定し(ステップ313)、また、「書き換え」が出力されていれば、ロック状態情報部2x4に、「2(ライトロック状態)」を設定して

(ステップ314)、次の図5に示す処理を行なう。
 【0021】以下、図5における処理動作の説明を行なう。図2のリード/ライトバス33に、アクセス種別として「書き換え」ではなく「読み出し」が出力されていれば(ステップ315)、エントリEのモディファイ部を「0(ブロックデータ部のBaとメインメモリのBaが一致)」とすると共に、エントリEのブロックデータ部に格納されているブロックBaのアドレスaで指定されるデータを、図2のデータバス32に出力する(ステップ316)。また、ステップ315において、図2のリード/ライトバス33に、アクセス種別として「書き換え」が出力されていれば、エントリEのモディファイ部2x6を、「1(ブロックデータ部のBaとメインメモリのBaが不一致)」とすると共に、エントリEのブロックデータ部に格納されているブロックBaのアドレスaで指定されるデータを、図2のデータバス32に出力されている内容で書き換える(ステップ317)。以上の処理が終了すると、図2のキャッシュメモリ2は、図2の転送応答バス36を介して、図2の処理ユニット1に、図2のデータバス32に読み出しデータを出力した旨、または、書き換えアクセスが完了した旨を通知する(ステップ318)。

【0022】また、図3のステップ303において、図2の処理ユニット1から、図2のアドレスバス31を介して要求されたアドレスaのデータを含むブロックBaを格納する有効なエントリEが、図2のキャッシュメモリ2に存在する場合、すなわち、図1における信号線21B~2eBの出力が「オン」であるエントリが存在する場合には、次の図6~図8で示す処理動作を行なう。

【0023】まず、図6における処理動作の説明を行なう。本図6は、エントリEのロック情報情報部2x4が(ステップ319)、「0(アンロック状態)」の場合の処理を示している。図2の処理ユニット1が、図2のロック指示バス35にロック指示信号を出力し(ステップ320)、かつ、図2のリード/ライトバス33に、アクセス種別として、「書き換え」を出力しているとき(ステップ321)、エントリEのモディファイ部2x6が、「1(ブロックデータ部のBaとメインメモリのBaが不一致)」であれば、図2のキャッシュメモリ2

は、図2のメモリバス5を介して、図2のメインメモリ4のブロックBaを、ブロックデータ部2x2の内容で書き換えると共に、モディファイ部2x6を「0(ブロックデータ部のBaとメインメモリのBaが一致)」とする(ステップ322)。

【0024】そして、ロック状態情報部2x4に「2(書き換え/ライトロック状態)」を設定し(ステップ323)、タスク識別情報部2x3に、図2のタスク識別情報バス34に出力されている値を設定する(ステップ324)。また、ステップ321において、アクセス種別として、「書き換え」ではなく「読み出し」を出力しているときは、ロック状態情報部2x4に「1(読み出しロック状態)」を設定し(ステップ325)、ステップ324の処理を行なう。このステップ324の処理のあとには、上述の図5に示す処理動作を行なう。

【0025】次に、図7における処理動作の説明を行なう。本図7は、図6のステップ319において、エントリEのロック情報情報部2x4が「1(リードロック状態)」の場合の処理を示している。図2の処理ユニット1が、図2のロック指示バス35に、ロック指示信号を出力し(ステップ326)、かつ、図2のリード/ライトバス33にアクセス種別として「書き換え」を出力しているとき(ステップ327)、図2のタスク識別情報バス34の全ての信号線の値と、エントリEのタスク識別情報部2x3の対応する挿げのビット値とが一致しなければ、すなわち、図1の信号線2xcの出力がオフであれば(ステップ328)、後述の図8におけるステップ329の処理を行なう。すなわち、図2のキャッシュメモリ2は、図2の矛盾応答バス38を介して、矛盾したアクセス、すなわち、他のタスクがリードロック中のブロックに対して、書き換えアクセスの要求を行なった旨を通知し、当該アクセスを完了する。

【0026】また、ステップ328において、図2のタスク識別情報バス34の全ての信号線の値と、エントリEのタスク識別情報部2x3の対応する挿げのビット値とが一致、すなわち、図1の信号線2xcの出力がオンであり、かつ、エントリEのモディファイ部2x6が「1(ブロックデータ部のBaとメインメモリのBaが不一致)」であれば、図2のキャッシュメモリ2は、図2のメモリバス5を介して、図2のメインメモリ4のブロックBaを、ブロックデータ部2x2の内容で書き換え、モディファイ部2x6を「0(ブロックデータ部のBaとメインメモリのBaが一致)」に設定する(ステップ330)。そして、ロック状態情報部2x4に「2(書き換えロック状態)」を設定する(ステップ331)。この場合、図2のタスク識別情報バス34の出力とエントリEのタスク識別情報部2x3の値とが等しく、ブロックBeをリードロック状態としたのは、処理ユニットで実行中のタスクのみである。

【0027】また、ステップ327において、図2のリ

ード/ライトバス33に、アクセス種別として「読み出し」を出力していれば、タスク識別情報部2x3に、タスク識別情報部2x3の値とタスク識別情報バス34に出力されている値との論理和を設定する(ステップ332)。例えば、タスク識別情報部2x3に、値「01101010」が設定されており、タスク識別情報バス34に、値「00000100」が出力されている場合、タスク識別情報部2x3には、値「01101110」が設定される。

【0028】また、ステップ326において、図2のロック指示バス35にロック指示信号を出力していない場合、タスク識別情報部2x3に、タスク識別情報部2x3の値と、タスク識別情報バス34に出力されている「0」と「1」を反転した値との論理積を設定する(ステップ333)。例えば、タスク識別情報部2x3に値「01101110」が設定されており、タスク識別情報バス34に値「00000100」が出力されている場合、タスク識別情報部2x3には、「値「01101010」が設定される。そして、タスク識別情報部2x3の全てのビットが「0」(例えば、「00000000」)となったときには(ステップ334)、ロック状態情報部2x4に、値「0(アンロック状態)」を設定する(ステップ335)。このような処理のあとには、上述の図5に示す処理動作を行なう。

【0029】次に、図8における処理動作の説明を行なう。本図8は、図6のステップ319において、エントリEのロック情報情報部2x4が「2(ライトロック状態)」の場合の処理を示している。図2の処理ユニット1が、図2のタスク識別情報バス34に出力している内容と、エントリEのタスク識別情報部2x3の値とが等しくなければ、すなわち、図1における信号線2xcの出力がオフであれば(ステップ336)、図2のキャッシュメモリ2は、図2の矛盾応答バス38を介して、矛盾したアクセス、すなわち、他のタスクがリードロック中のブロックに対して、書き換えアクセスの要求を行なった旨を通知し(ステップ329)、当該アクセスを完了する。また、ステップ336において、図2のタスク識別情報バス34の出力と、エントリEのタスク識別情報部2x3の値とが等しく、かつ、図2の処理ユニット1が、図2のロック指示バス35にロック指示信号を出力していない場合(ステップ337)には、タスク識別情報部2x3の全てのビットに「0」を設定し、ロック状態情報部2x4に「0(アンロック状態)」を設定する(ステップ338)。そして、以下、上述の図5に示す処理動作を行なう。

【0030】尚、ステップ329における矛盾したアクセスを行なった旨の通知を受けると、図2の処理ユニット1は、実行中のタスクを中断して、予め定められた時間が経過したあとに、タスクを再開するか、もしくは、実行中のタスクが書き換えたデータの内容を、タスクが

実行を開始する時点の状態に戻す処理を行なって、実行中の処理を無効化(中止)し、予め定められた時間が経過したあとに処理を最初から実行する。このような、実行中のタスクの処理の無効化(中止)は、無効化する図1におけるタスク識別情報部213~2e3の値が等しく、かつ、図1のモディファイ部216~2e6に、値「1(ブロックデータ部のBaとメインメモリのBaが不一致)」が設定されている全てのエントリのパリッド部215~2e5に、値「0(無効)」を設定することによって行なうことができる。

【0031】無効化するタスク識別情報と、図1のタスク識別情報部213~2e3の値が等しく、かつ、図1のモディファイ部216~2e6に、値「1(ブロックデータ部のBaとメインメモリのBaが不一致)」が設定されているエントリに格納されているブロックの内容は、無効化するタスクによって書き換えられる前に、図2のメインメモリ4に書き込まれているので、上述の処理を終了したあと、他のタスクは、書き換えられる前のブロックにアクセス可能となる。

【0032】また、必ずしも、図2の矛盾応答バス37を介して矛盾が発生した旨の通知を受信したときに実行中のタスクを中断、または、無効化せず、図2の処理ユニット1が、原因となった当該ブロックを格納しているエントリのタスク識別情報部の内容を読みだす処理部を設け、当該ブロックをロックしているタスクを無効化するものでも良い。これにより、共通データに並列にアクセスするタスクの中で、重要度が高く、他のタスクに比べて優先して実行させたい優先度の高いタスクと、優先度の低いタスクとの間で、共通データに対して競合が発生した場合、予め、タスクごとに割り当てられた優先度によって、優先度の低い処理を中止させ、優先度の高い処理を優先して実行させることが可能となる。

【0033】以上のように、図1に示す実施例のキャッシュメモリでは、処理ユニットで実行するタスクを相互に切り替えて実行するマルチタスク処理において、複数のタスクが並列にアクセスする共通データのアクセス権の確保と開放(ロックとアンロック)の処理を、各タスクが共通データにアクセスする処理と同時に行なうことができる。このことにより、共通データのアクセス権の確保と開放のためのタスクの処理量の増加が生じず、また、タスクの切り替えを抑止することもなく、排他制御のための実効的な処理性能の低下をなくすることが可能となる。

【0034】上述の実施例において、図2の処理ユニット1で実行中のタスクからの共通データへのアクセスが不要となったとき、図2の処理ユニット1は、図2のキャッシュメモリ2に対して、当該タスクのタスク識別情報とロック状態の解除を指示する。このような指示が入力されると、図1のキャッシュメモリ2は、次の図9で示すロック解除の処理を行なう。

【0035】図9は、図2における情報処理システムの本発明に係わる処理動作の第2の具体例を示すフローチャートである。本例は、図2の処理ユニット1からのロック解除指示に対する図1のキャッシュメモリ2のロック解除の処理動作を示すものであり、このキャッシュメモリ2内には、エントリを順次指定可能なカウンタを設けている。そして、論理回路を用いることにより、この処理の各部分を並列に実行することができる。図1のキャッシュメモリ2は、図2のキャッシュバス3を介して、ロック解除の指示（予め定められたアドレスへのアクセス）と共に、タスク識別情報バス34を介して、タスク識別情報（このタスク識別情報と共に出力されたロック指示によって設定された全てのエントリがロック解除処理の対象となる）を受信すると、エントリを順次指定可能なカウンタに初期値（ここでは「1」）を設定する（ステップ401）。このカウンタ値が図1のキャッシュメモリ2内のエントリ総数eになるまで、カウンタの値が+1されるごとに、以下の処理を行なう。尚、ここで、カウンタの値「c」で指定されるエントリをエン

トリEcとし、このエントリEcは、アドレス情報部2c1、ブロックデータ部2c2、タスク識別情報部2c3、・・・で構成されているものとする。

【0036】エントリEcのバリッド部2c5の値が「0（無効）」（ステップ402）、または、ロック状態情報部2c4の値が「0（アンロック状態）」の場合（ステップ403、404）、カウンタの値を+1する（ステップ410）。ステップ402において、エントリEcのバリッド部2c5の値が「1（有効）」で、かつ、ステップ403、404において、ロック状態情報部2c4の値が「1（リードロック状態）」の場合には、タスク識別情報部2x3に、タスク識別情報部2x3の値と、タスク識別情報バス34に出力されている「0」と「1」を反転した値との論理積を設定する（ステップ405）。例えば、タスク識別情報部2x3に値「01101110」が設定されており、タスク識別情報34に値「00000100」が出力されている場合、タスク識別情報部2x3には、値「01101010」が設定される。タスク識別情報部2x3の全てのビットが「0」（例えば、「00000000」）となったときには（ステップ406）、ロック状態情報部2x4に値「0（アンロック状態）」を設定し（ステップ407）、カウンタの値を+1する（ステップ410）。

【0037】また、ステップ402において、エントリEcのバリッド部2c5の値が「1（有効）」で、ステップ403において、ロック状態情報部2c4の値が「2（ライトロック状態）」の場合で、かつ、エントリEcのタスク識別情報部2c3とタスク識別情報バス34の値が等しい場合、すなわち、信号線2cCの値がオンの場合（ステップ408）、バリッド部2c5に値「0（無効）」をセットし（ステップ409）、リード

／ライトバス33の値を+1する（ステップ4109）。ステップ408において、エントリEcのタスク識別情報部2c3とタスク識別情報バス34の値が等しくない場合、すなわち、信号線2cCの値がオフの場合、そのまま、カウンタの値を+1する（ステップ410）。

【0038】このようにして、図2の処理ユニット1で実行するタスクにおいて、リードロック状態、または、ライトロック状態となっている共通データへのアクセスが不要となった場合、タスクが設定した排他制御情報を解除するためのアクセス（すなわち、ロック指示を行なわずに、当該ブロックへのアクセスを行なう）を、1回で行なうことができ、従来技術のように、共通データを格納しているブロックごとに行なうことが不要となり、ロック解除のための処理ユニットからのアクセス回数を削減することができる。特に、矛盾発生のお知らせを受信した場合の無効化のために、全てのロック状態を解除する場合に有効である。また、ロック状態の解除の指示と共に通知するタスク識別情報で、全てのタスクを指定する（例えば、「11111111」を出力）ことによって、図1におけるキャッシュメモリ2の全てのロック情報を解除することも可能となる。

【0039】以上、図1～図9を用いて説明したように、本実施例のキャッシュメモリでは、タスクが並列にアクセスする共通データへのアクセスを行なう際に、アクセス権の確保と開放の処理を、各処理ユニットで実行するアクセスと同時に実行することができる。このことにより、タスク間でアクセスするデータの無矛盾性を、効率良く保証することが可能である。また、タスクの並列処理を阻害する無用な排他制御、および、タスクの切り替えを抑止する制御などが不要となり、アクセス権の確保と開放のための処理ユニットのオーバヘッドも生じず、処理ユニットでの実効的な性能を向上させることが可能となる。尚、本発明は、図1～図9を用いて説明した実施例に限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能である。

【0040】

【発明の効果】本発明によれば、キャッシュメモリのエントリに格納されているブロックへの各タスクのアクセスに対する排他制御を効率良く行なうことができ、タスク間でアクセスするデータの無矛盾性の保証処理の高効率化と、マルチタスク処理を行なう処理ユニットの実効的な性能の向上が可能である。

【図面の簡単な説明】

【図1】本発明の排他制御システムに用いるキャッシュメモリの本発明に係わる構成の一実施例を示すブロック図である。

【図2】図1におけるキャッシュメモリを用いた情報処理システムの構成の一具体例を示すブロック図である。

【図3】図2における情報処理システムの本発明に係わる処理動作の第1の具体例を示すフローチャートの第1

の部分である。

【図4】図2における情報処理システムの本発明に係わる処理動作の第1の具体例を示すフローチャートの第2の部分である。

【図5】図2における情報処理システムの本発明に係わる処理動作の第1の具体例を示すフローチャートの第3の部分である。

【図6】図2における情報処理システムの本発明に係わる処理動作の第1の具体例を示すフローチャートの第4の部分である。

【図7】図2における情報処理システムの本発明に係わる処理動作の第1の具体例を示すフローチャートの第5の部分である。

【図8】図2における情報処理システムの本発明に係わる処理動作の第1の具体例を示すフローチャートの第6の部分である。

【図9】図2における情報処理システムの本発明に係わる処理動作の第2の具体例を示すフローチャートである。

【符号の説明】

1 処理ユニット

2 キャッシュメモリ

3 キャッシュバス

4 メインメモリ

5 メモリバス

21、22 信号線

31 アドレスバス

32 データバス

33 リード/ライトバス

34 タスク識別情報バス

35 ロック指示バス

10 36 転送応答バス

37 矛盾応答バス

210～2e0 エントリ

211～2e1 アドレス情報部

212～2e2 ブロックデータ部

213～2e3 タスク識別情報部

214～2e4 ロック状態情報部

215～2e5 バリッド部

216～2e6 モディファイ部

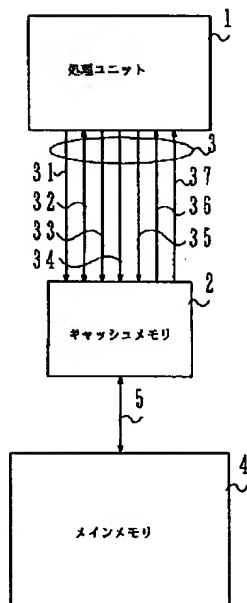
217～2e7 比較器

20 218～2e8 比較器

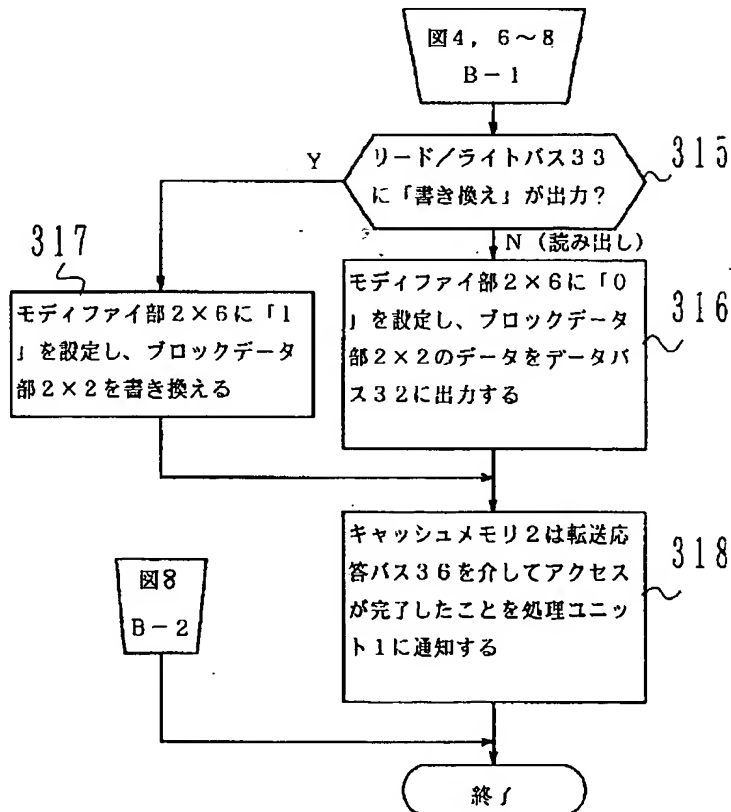
219～2e9 アンドゲート

21B～2eB 信号線

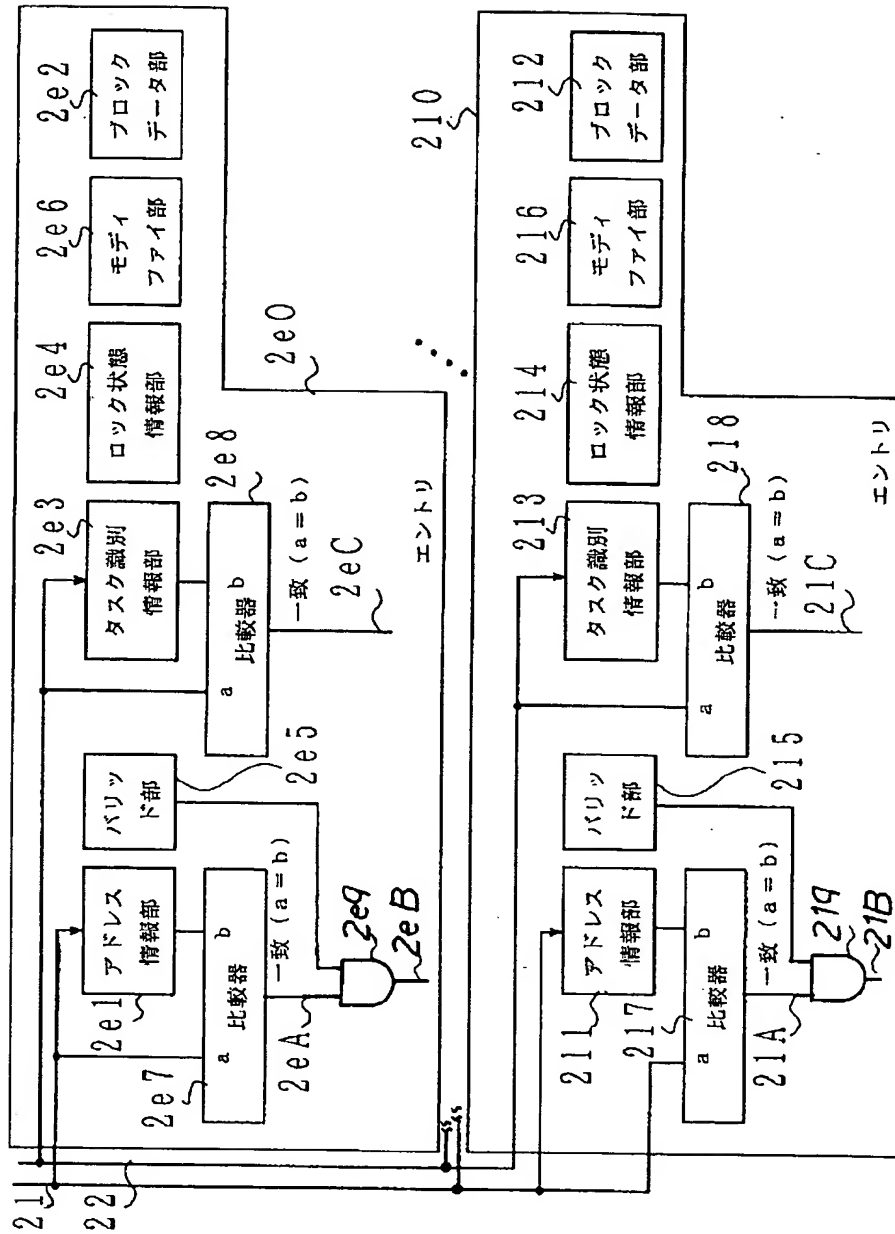
【図2】



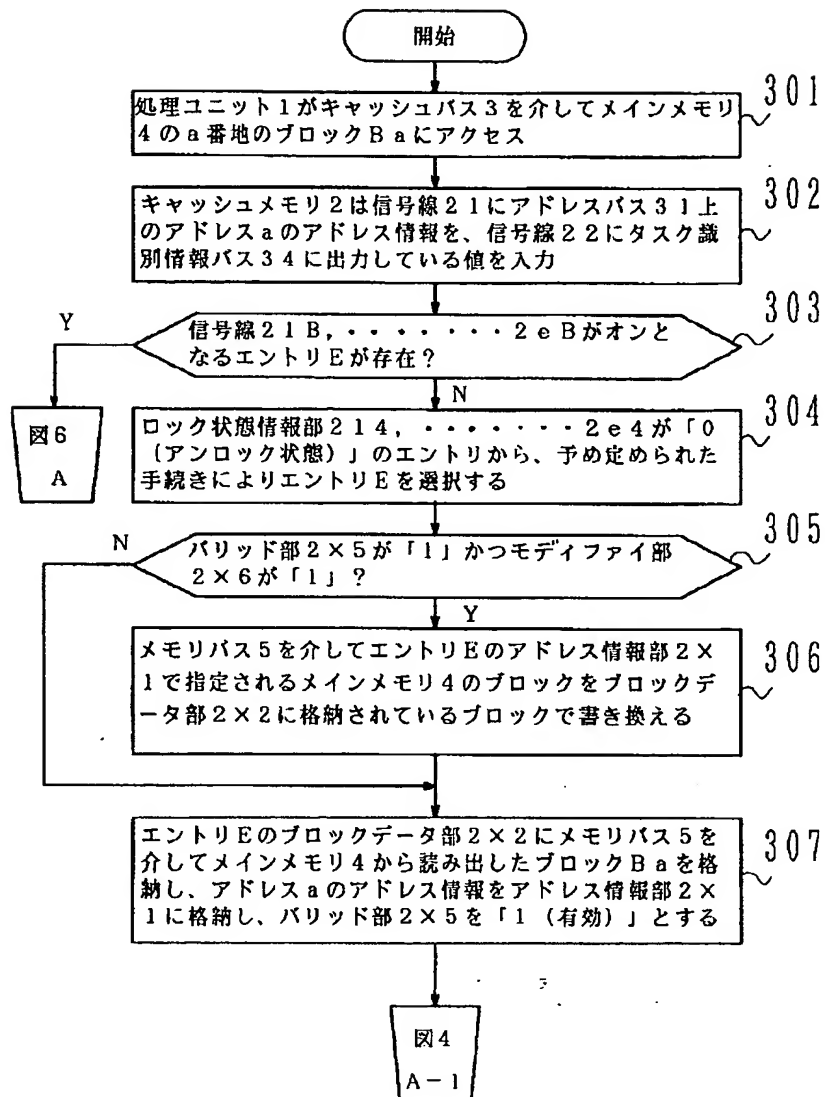
【図5】



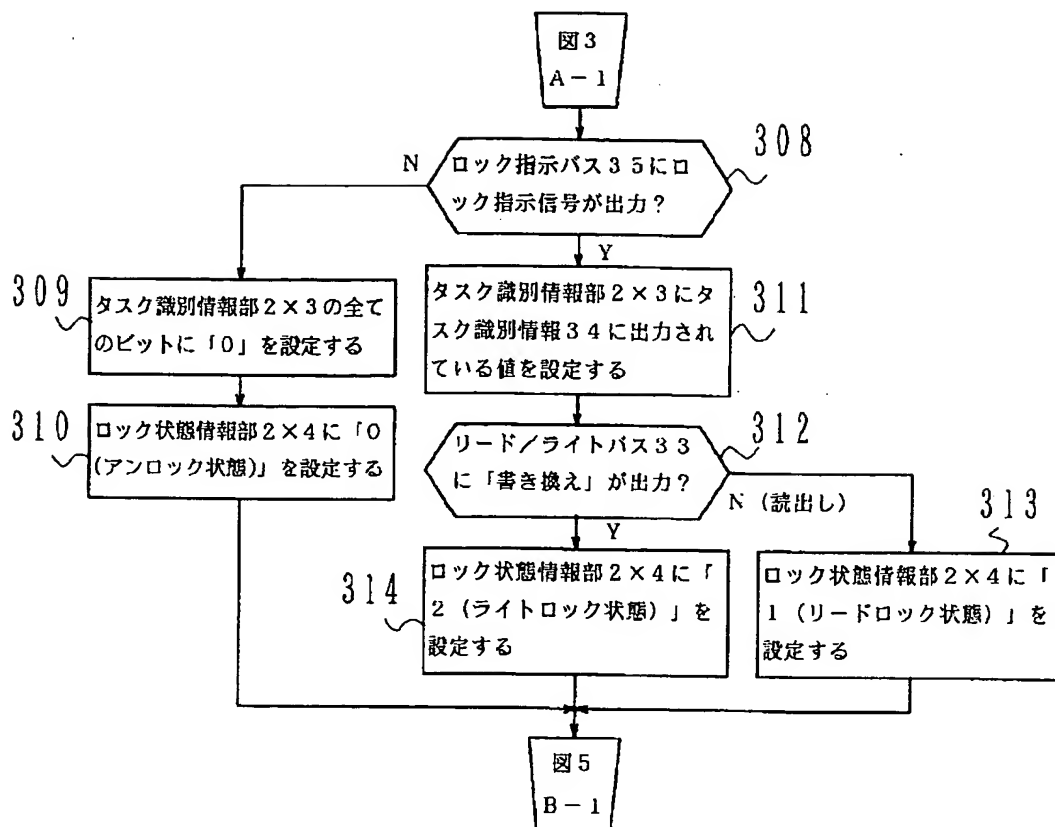
【図1】



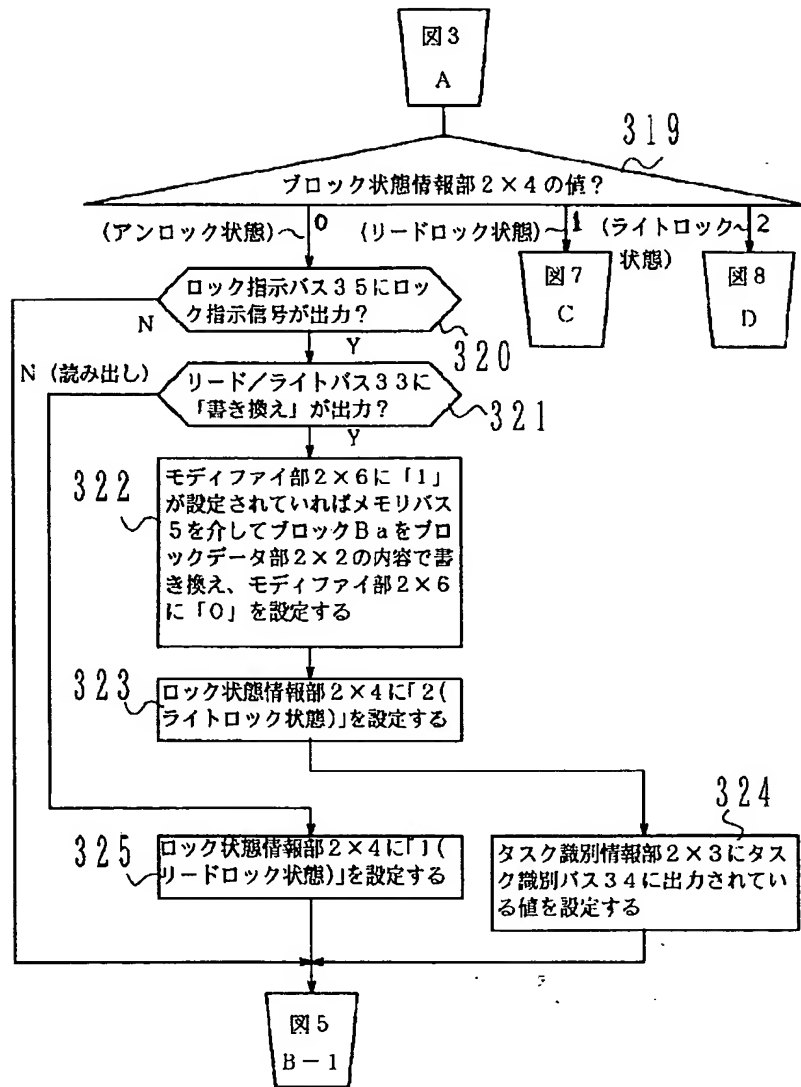
【図3】



【図4】

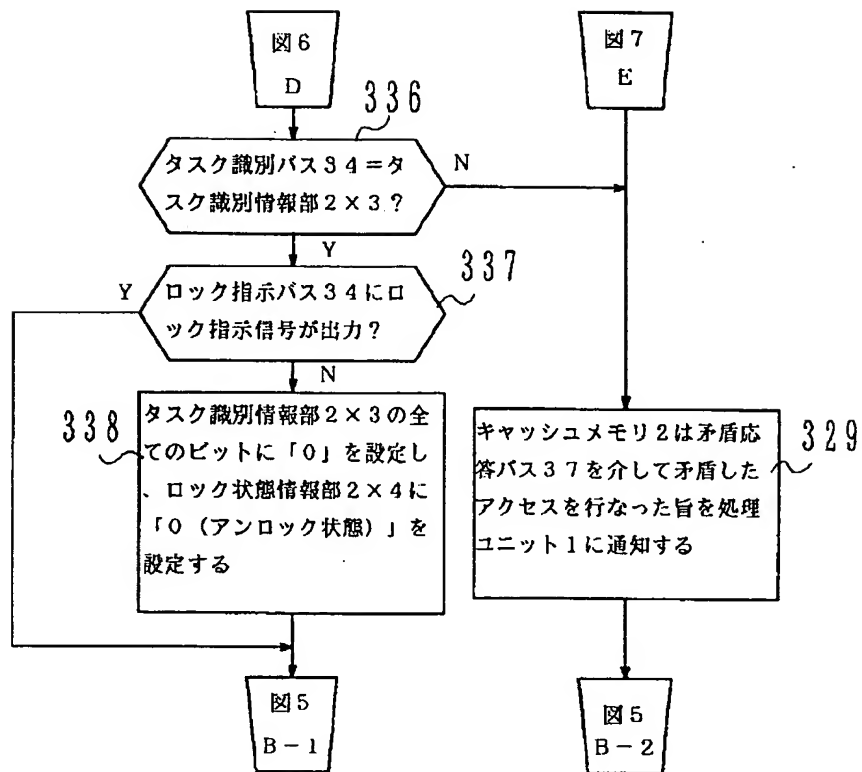


【図6】



[illegible]

【図8】



【図9】

